Lecture 22

plan: 1) Finish matroid union
2) Ellipsoid

Ellipsoid Algorithm

- general purpose convex optimization algorithm
- polynomial time in many situations, e.g. linear programming.
- Slow for LP in practice

- Contrast w/ simplex, which is <u>fast</u> in practice, but not provably polynomial.
- Ellipsoid has many complexity consequences in optimization.
- There's another class of algos, <u>interior point methods</u>
  e.g. Karmarkar's algorithm '84
  which solve LP fast in practice <u>and</u> in theory but are not as theoretically consequential.

Consequences

Given convex set $P \subseteq \mathbb{R}^n$, (e.g. a polyhedron), we study two problems:

- **Separation (SEP):**

  Given $y \in \mathbb{R}^n$, decide if $y \in P$, & if not return **separating hyperplane**, i.e. $c \in \mathbb{R}^n$ s.t. $c^T y > \max\{c^T x : x \in P\}$

- **Optimization (OPT)**

  Given vector $c \in \mathbb{R}^n$, find x maximizing $c^T x$ on P.

Examples

- **Linear programming:**

  if $P = \{x : Ax \leq b\}$, can solve SEP by just checking $a_i^T x \leq b_i$ for rows a_i^T of A;

returning $c = a_i^T$ if violated.

efficient if A is part of input.

- For $P = \{x : Ax \leq b\}$, <u>OPT</u> is just linear programming:

$$\max c^T x$$
$$\text{subject to } Ax \leq b.$$

SEP EZ, OPT seems hard.

- <u>Matroid polytope</u>:

$M = (E, \mathcal{I})$ matroid,

$P = \mathrm{conv}(\{1_S : S \in \mathcal{I}\})$

we know face characterization:

Thm

$$\boxed{P = \{x \in \mathbb{R}^E : \begin{array}{ll} x(S) \le r(S) & \forall S \subseteq E \\ x_e \ge 0 & \forall e \in E \} \end{array}}.$$

However, exponentially many constraints!

So even if we can

efficiently compute r,
SEP not obvious!

- OPT for P is just max cost independent set, efficiently solvable by greedy (provided we can decide if $S \in \mathcal{I}$ efficiently).

  OPT EZ, SEP seems hard!

- Matroid intersection polytope:

  OPT?? SEP??

- **Amazing Result:**

Theorem (Grötschel, Lovász, Schrijver '81) For a family of convex sets P,

$$\boxed{\begin{array}{c}\text{SEP for } P \text{ poly-time solvable} \\ \Longleftrightarrow \\ \text{OPT for } P \text{ poly-time solvable.}\end{array}}$$

Proof idea:

$\Rightarrow$ Ellipsoid algorithm: can optimize over P using only calls to SEP oracle.

($\Leftarrow$) Reduce to ($\Rightarrow$) using notion called "polar" P^* of P; we won't cover this.

- Actually, if P is not "too degenerate", don't even need SEP, just MEM! decide if $x \in K$.

Thm (GLS '88): If P contains ball of radius ε & P contained in ball of radius R, can implement SEP with poly$(\log(\frac{1}{\varepsilon}), \log(R), n)$ calls to MEM.*

Actually, is about approximate versions of SEP & MEM.

Proof: not covered.

OPT vs. feasibility

- First we solve simpler problem:

$\boxed{\text{FEAS: Given SEP oracle for } P \text{, find some } x \in P \text{ or decide } P = \emptyset.}$

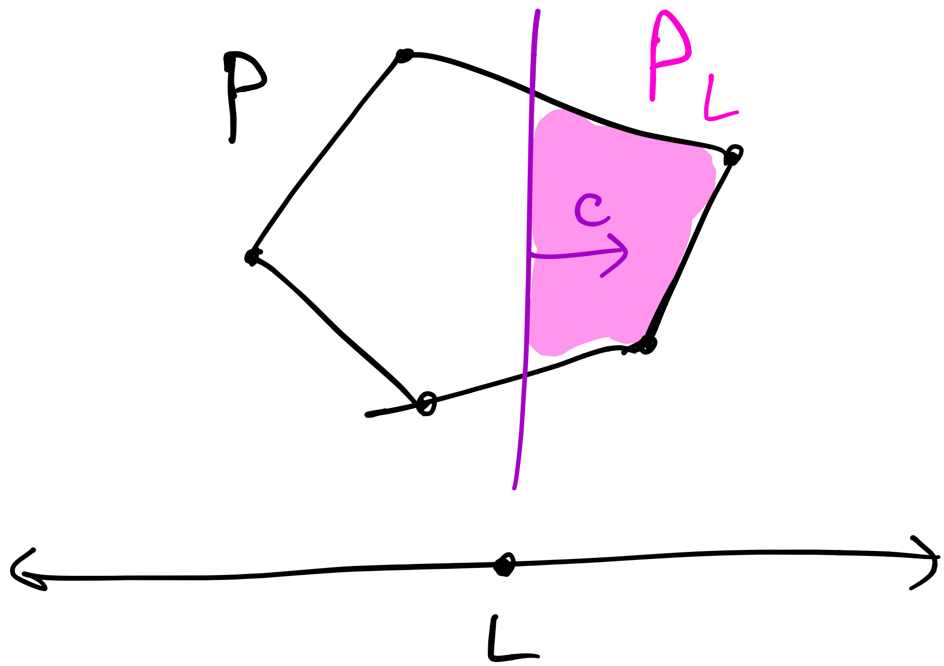
- OPT reduces to FEAS:

binary search:

$\max\{c^T x : x \in P\} \geq L$ if

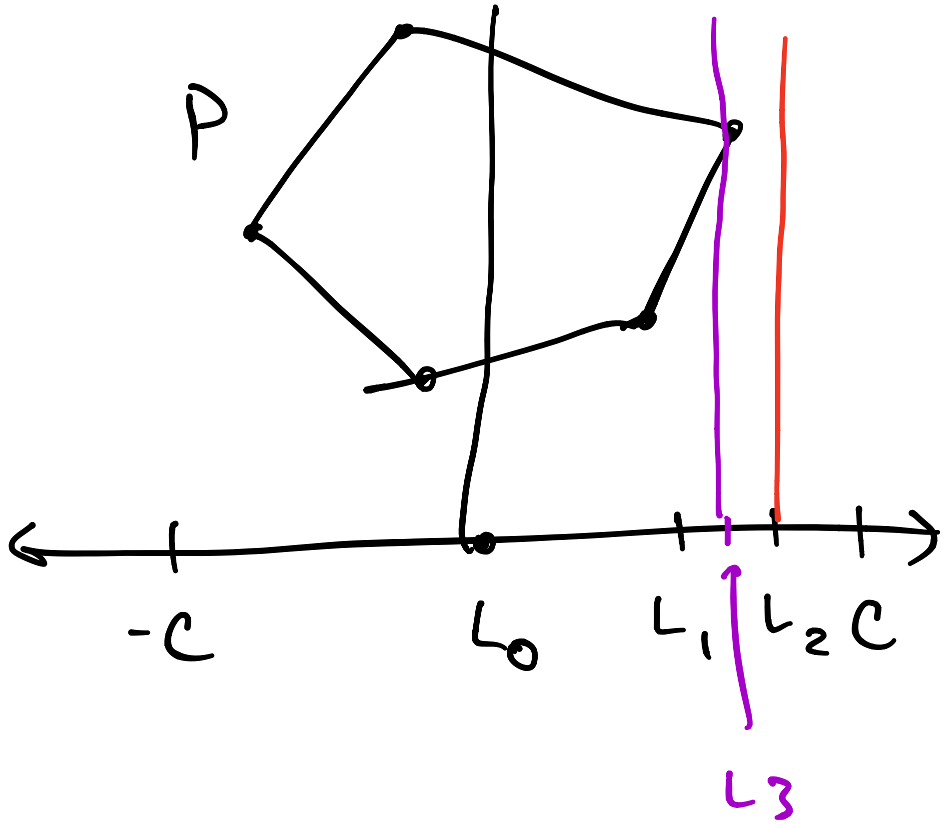
$P_L := P \cap \{x : c^T x \geq L\} \neq \emptyset.$

so just "search" for max L s.t. $P_L \neq \emptyset$.



- Given a-priori bound $-C \leq L \leq C$,

Binary search to find max L s.t. P_L feasible.



- optimizes to ε- precision in $\log(C/\varepsilon)$ time.
- for LP, can solve exact opt with $C, 1/\varepsilon \leq$ exponential in bitsize of A, b.

Details later for special case.

(finally!)

The Algorithm

- Solves FEAS in time $\text{poly}(\log(\frac{1}{\varepsilon}), \log(R), n)$ as in GLS 88 (still need small/large ball condition on P).

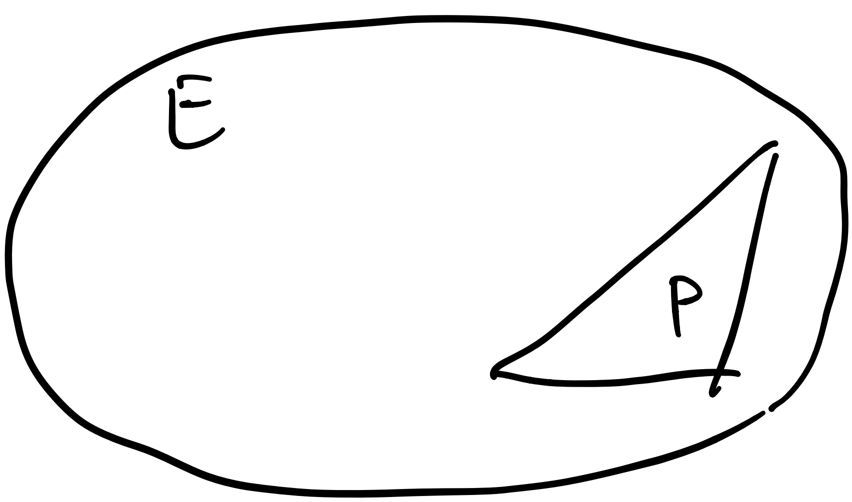
- ε, R dependence not a big deal: (& actually necessary).

For LP with $K = \{x : Ax \leq b\}$, ε, R can be assumed $\text{poly}(\text{total bitsize of } A, b)$.

using some tricks. We'll see these tricks later in special case.

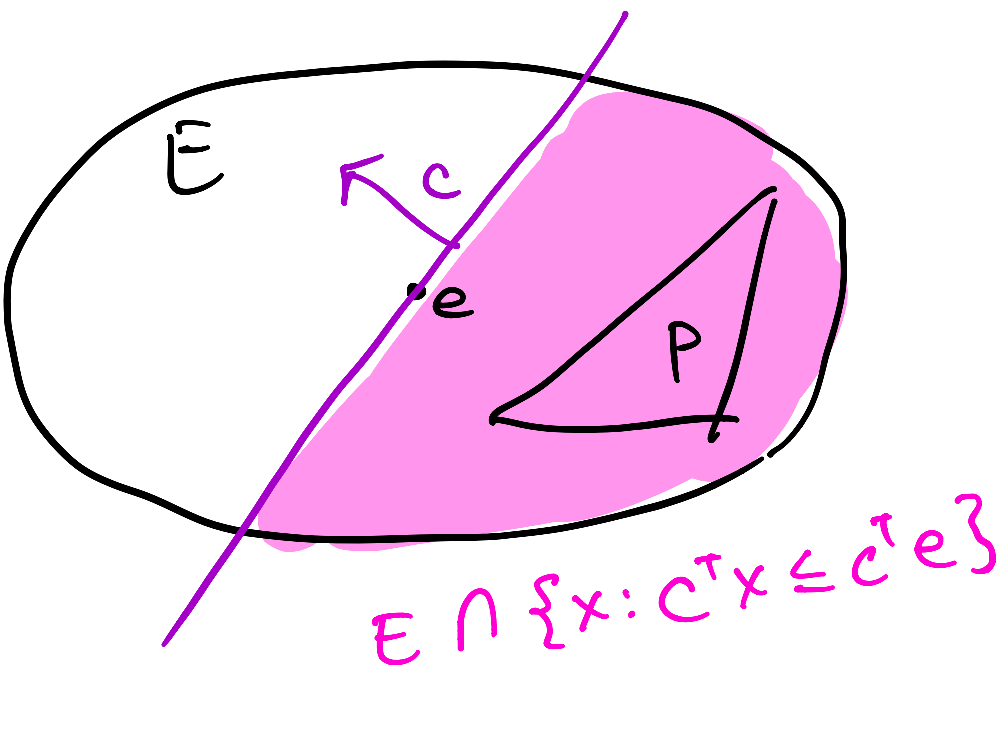
Algorithm idea:

- Set $E = E_0$, E_0 ellipsoid guaranteed to contain P.
  (Can find E_0 b/c given ball containing P).

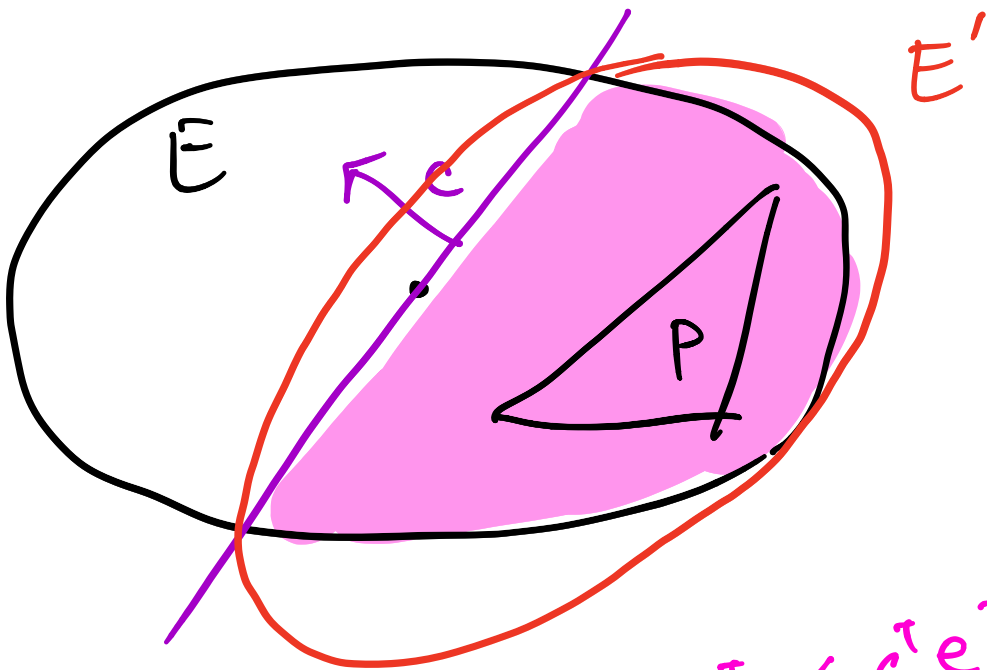


- While not done:

- Check if center e of E is in P. (using call to SEP).
  - if so, return e.
  - if not, SEP outputs separating hyperplane $c^T x \leq d$. (actually, can assume $d = c^T e$)



▷ Let E' "smaller" ellipse containing $E \cap \{x : c^T x \le c^T e\}$



$E \cap \{x : c^T x \le c^T e\}$

Actually, take E' to be minimum volume ellipse containing $E \cap \{x : c^T x \le c^T e\}$.

▷ Set $E \leftarrow E'$.

Runtime:

- **Volume lemma:**

$$\mathrm{vol}(E') \leq e^{-\frac{1}{2(n+1)}} \mathrm{vol}(E).$$

- As E always contains P, alg. will terminate in

$$2(n+1) \log \left(\frac{\mathrm{vol}(E_0)}{\mathrm{vol}(P)} \right)$$

iterations.

Issues:

- How to compute E'?
- Proof of Volume Lemma?
- When to output that P is empty?